Introduction to Google Cloud

Before you start exploring the world of Kubernetes, let's take some time to introduce you to, or remind you of, some Google Cloud core concepts.

Introduction to Google Cloud

01 Google Cloud compute offerings

02 Google's global network

03 Managing cloud resources

04 Billing tools

05 Four ways to interact with Google Cloud

In this first section of this course, you explore:

- Cloud computing and the services that Google Cloud offers to architects and developers to build solutions.
- How Google's powerful global network can power Google Cloud services.
- How Google Cloud resources are structured and managed.
- Tools to ensure that your organization doesn't accidentally face a big Google Cloud bill.
- And four different ways to interact with Google Cloud to commence work.

You also get some hands-on practice accessing the Google Cloud console and Cloud Shell.

Introduction to Google Cloud

01 Google Cloud compute offerings

02 Google's global network

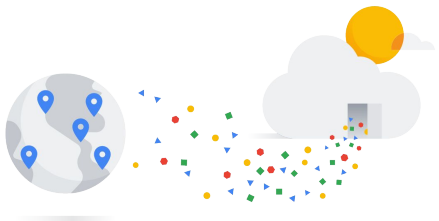
03 Managing cloud resources

04 Billing tools

05 Four ways to interact with Google Cloud

Let's start with an overview of cloud computing and the different Google Cloud compute offerings available.

Five traits of cloud computing



01 Customers get computing resources that are on-demand and self-service.

02 Customers get access to those resources over the internet, from anywhere.

03 The provider of those resources allocates them to users out of that pool.

04 Resources are elastic, which means they're flexible, so customers can be.

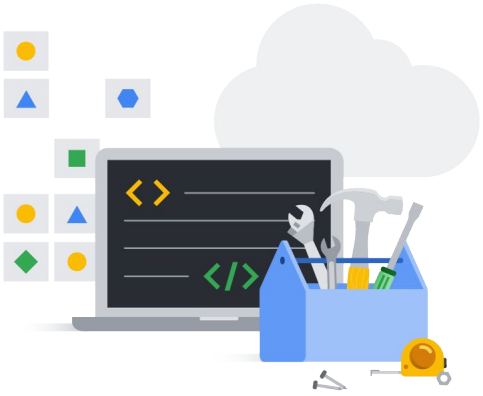
05 Customers pay only for what they use or reserve as they go.

Cloud computing is a way of using information technology, IT, that has these five equally important traits.

1. First, customers get computing resources that are on-demand and self-service. Through a web interface, users get the processing power, storage, and network they require, without the need for human intervention.
2. Second, customers get access to those resources over the internet from anywhere they have a connection.
3. Third, the cloud provider has a big pool of those resources and allocates them to users out of that pool. That allows the provider to buy in bulk and pass the savings on to the customers. Customers don't have to know or care about the exact physical location of those resources.
4. Fourth, the resources are elastic–which means they're flexible–so customers can be. If customers need more resources they can get more and quickly. If they need less, they can scale back.
5. And finally, customers pay only for what they use, or reserve as they go. If they stop using resources, they stop paying.

That's it. That's the definition of cloud.

Google Cloud services to build solutions



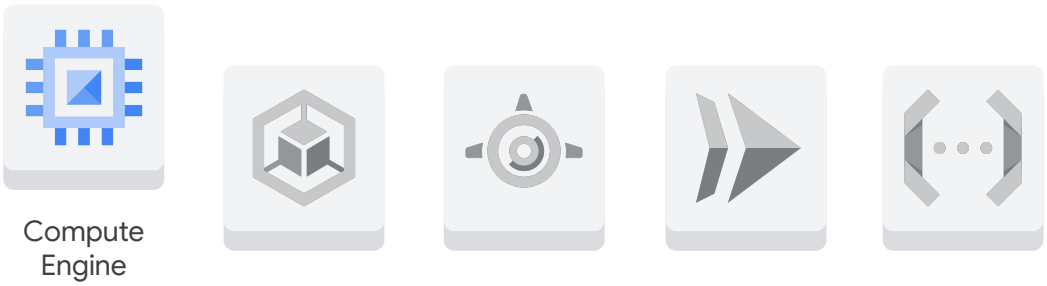
Virtual machines

Google Kubernetes Engine

Google Cloud offers a variety of services for architects and developers to use to build solutions.

Some might sound familiar, like virtual machines, whereas others might represent a totally new paradigm, like Google Kubernetes Engine.

Google Cloud computing services



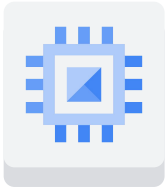
Let's start with **Compute Engine**.

Compute Engine

- IaaS offering
- Predefined and customized VMS
- Persistent disks and local SSDs
- Managed instance groups
- Per-second billing

- Compute Engine is an IaaS offering, or infrastructure as a service, which provides compute, storage, and network virtually that are similar to physical data centers.
- Compute Engine provides access to predefined and customized virtual machine configurations. At the time this training was developed, VMs could be as large as 416 vCPUs with more than 12 TB of memory.
- Virtual machines require block storage, and Compute Engine offers two main choices: Persistent disks and local SSDs. Persistent disks offer network storage that can scale up to 257 TB, and can disk snapshots for backup and mobility. Alternatively, local SSDs enable high input/output operations per second.
- Compute Engine workloads can be placed behind global load balancers that support autoscaling. They offer a feature called managed instance groups with which resources can be defined to automatically deploy to meet demand.
- Compute Engine costs can be controlled with the help of per-second billing. This means that when compute resources are deployed for short periods of time, like with batch processing jobs, costs can stay low. Compute Engine also offers spot virtual machines, which are preemptible and provide significantly cheaper pricing for workloads that can be interrupted safely.

Compute Engine is popular for developers



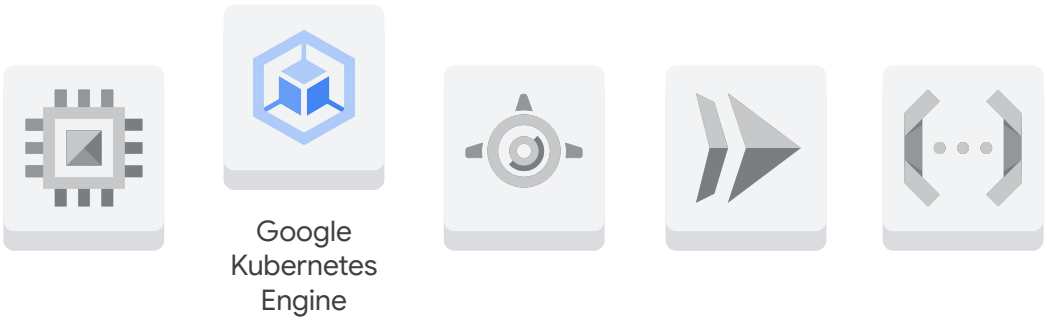
Compute Engine

- It provides complete control over infrastructure, because operating systems can be customized, and it can run applications that rely on a mix of operating systems.
- On-premises workloads can easily be lifted and shifted to Google Cloud without needing to rewrite.
- It's the best option when other computing options don't support your application or requirements.

Compute Engine is a popular choice for developers because:

- It provides complete control over infrastructure, because operating systems can be customized, and it can run applications that rely on a mix of operating systems.
- On-premises workloads can easily be lifted and shifted to Google Cloud without needing to rewrite applications or make any changes.
- It's the best option when other computing options don't support your application or requirements.

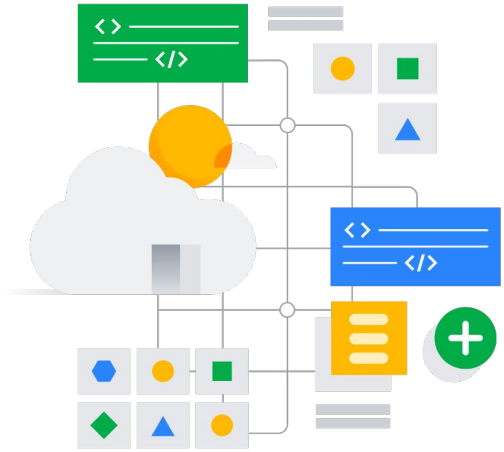
Google Cloud computing services



Google Cloud's second compute offering, and the focus of this course, is **Google Kubernetes Engine**.

Google Kubernetes Engine

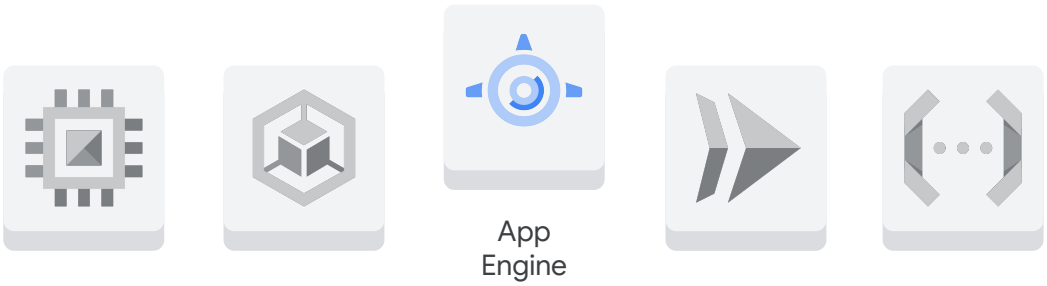
- Runs containerized applications.
- Contains code packaged with all its dependencies.



GKE runs containerized applications in a cloud environment, as opposed to on an individual virtual machine, like Compute Engine. A container represents code packaged with all its dependencies.

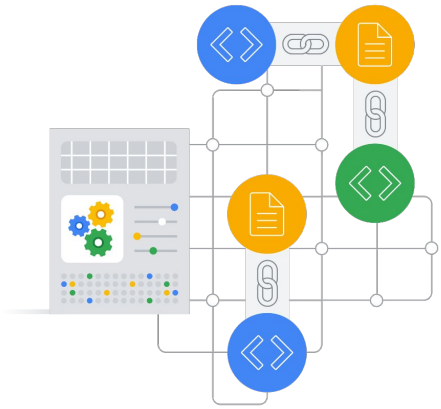
We get into more details about Kubernetes and containerized applications throughout this course. But for now, you can think of containerization as a way to package code so it's highly portable and resource-efficient. And Kubernetes is a way to orchestrate code in containers.

Google Cloud computing services



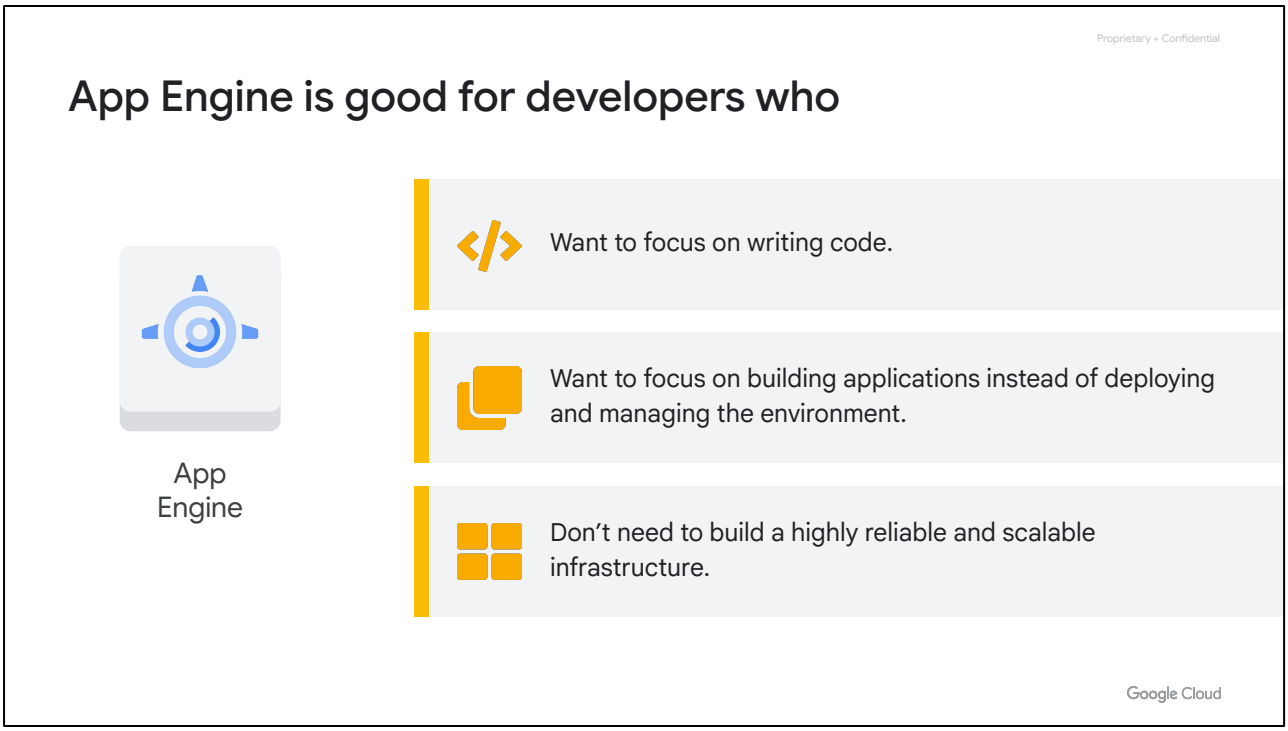
The third computing service offered by Google is **App Engine**.

App Engine



- Is a fully managed PaaS offering.
- Binds code to libraries.
- Focuses on application logic.
- Deploys required infrastructure.
- Integrates with Cloud tools.
- Offers version control and traffic splitting.

- App Engine is a fully managed PaaS offering, or platform as a service. PaaS offerings bind code to libraries that provide access to the infrastructure application needs. This allows more resources to be focused on application logic instead of deployment.
- This means developers can just upload code and App Engine will deploy the required infrastructure. It supports popular languages like Java, Node.js., Python, PHP, C#, .NET, Ruby, and Go, and can also be used to run container workloads.
- App Engine is closely integrated with Cloud Monitoring, Cloud Logging, Cloud Profiler, and Error Reporting.
- App Engine also supports version control and traffic splitting.



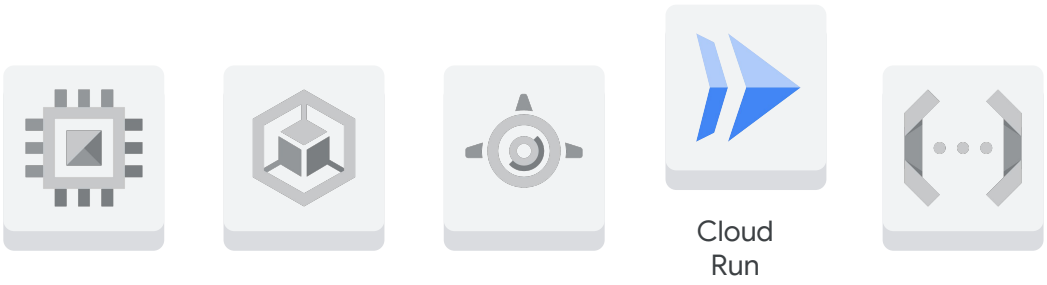
App Engine is a good choice for developers that:

- Want to focus on writing code.
- Want to focus on building applications instead of deploying and managing the environment.
- Don't need to build a highly reliable and scalable infrastructure.

Some of the most common App Engine use cases include:

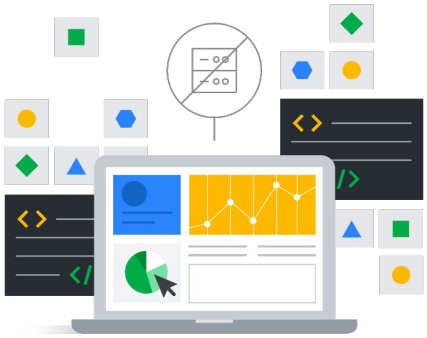
- Websites
- Mobile app and gaming backends
- And a method to present a RESTful API, which is an application program interface that resembles the way a web browser interacts with a web server, to the internet. App Engine makes them easy to operate.

Google Cloud computing services



Another compute option offered by Google Cloud is Cloud Run.

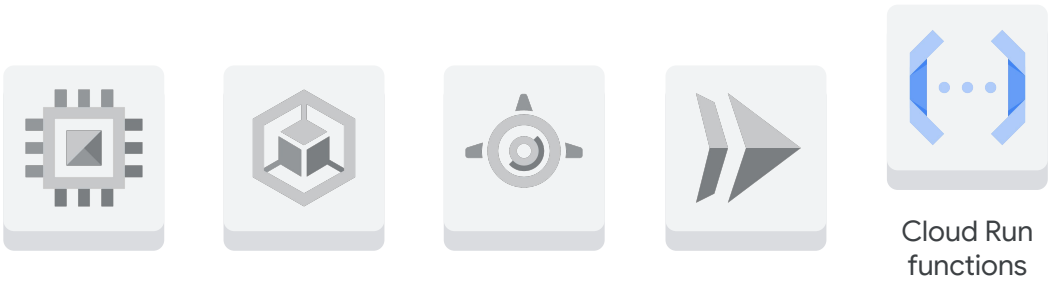
Cloud Run



- Is a managed compute platform.
- Runs stateless containers.
- Is serverless.
- Is built on Knative.
- Is fast.
- Charges only for resources used.

- Cloud Run is a managed compute platform that runs stateless containers through web requests or Pub/Sub events.
- Cloud Run is serverless. That means it removes all infrastructure management tasks so you can focus on developing applications.
- It's built on Knative, an open API and runtime environment that is built on Kubernetes and gives you freedom to move your workloads across different environments and platforms. It can be fully managed on Google Cloud, on Google Kubernetes Engine, or anywhere Knative runs.
- Cloud Run is fast. It can automatically scale up and down from zero almost instantaneously, and it charges only for the resources used, calculated down to the nearest 100 milliseconds, so you never pay for over-provisioned resources.

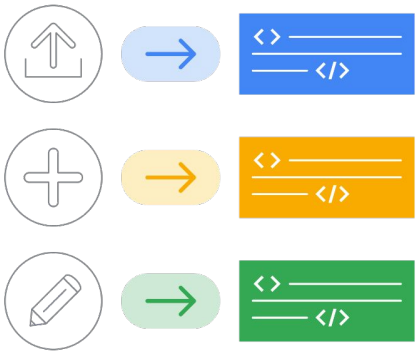
Google Cloud computing services



And finally, there is Cloud Run functions.

Cloud Run functions

- Is an event-based, asynchronous compute solution.
- Executes code in response to events.
- Functions as a service offering.
- Deploys the computing capacity to run code.
- Can connect and extend cloud services.
- Bills to the nearest 100 milliseconds.



- Cloud Run functions is a lightweight, event-based, asynchronous compute solution for creating small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment.
- It executes code in response to events, like when a new file is uploaded to Cloud Storage.
- It's also a completely serverless execution environment. Cloud Run functions is often referred to as functions as a service.
- Simply upload code written in Node.js, Python, Go, Java, .Net Core, Ruby, or PHP, and Google Cloud automatically deploys the appropriate computing capacity to run that code. These functions can be used to construct application workflows from individual business logic tasks.
- Cloud Run functions can also be used to connect and extend cloud services.
- You're billed to the nearest 100 milliseconds, but only while your code is running. Cloud Run functions also provides a perpetual free tier, so many Cloud Function use cases can be free of charge.

Common Cloud Run functions

Cloud Run functions

It can be part of a microservices application architecture.

It is used to build simple, serverless mobile or IoT backends or integrate with third-party services and APIs.

It can be part of intelligent applications such as virtual assistants, video or image analysis, and sentiment analysis.

What are common Cloud Run functions?

- It can be part of a microservices application architecture.
- It is used to build simple, serverless mobile or IoT backends or integrate with third-party services and APIs. Files uploaded to a Cloud Storage bucket can be processed in real time. Similarly, the data can be extracted, transformed, and loaded for querying and analysis.
- And it can be part of intelligent applications such as virtual assistants, video or image analysis, and sentiment analysis.

Introduction to Google Cloud

01 Google Cloud compute offerings

02 Google's global network

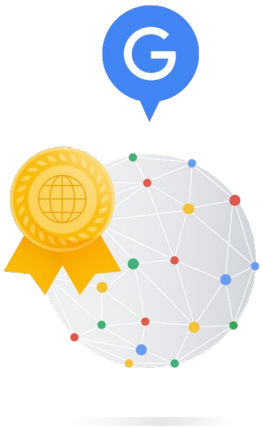
03 Managing cloud resources

04 Billing tools

05 Four ways to interact with Google Cloud

Behind the services provided by Google Cloud, lies a huge range of resources, from physical assets like servers, to virtual resources like virtual machines and containers. And it all runs on Google's own global network.

The Google network



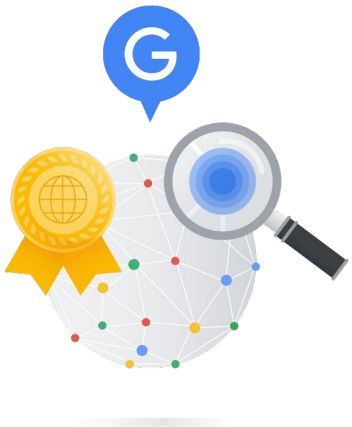
40%

of the world's internet traffic every day

According to some publicly available estimates, Google's network carries as much as 40% of the world's internet traffic every day. Google's network is the largest network of its kind, and Google has invested billions of dollars over the years to build it.

The Google network

- It carries as much as 40% of the world's traffic.
- It's designed to give the highest throughput.
- It's designed for the lowest possible latencies.
- It uses 100+ content caching nodes worldwide.
- High demand content is cached for quicker access.



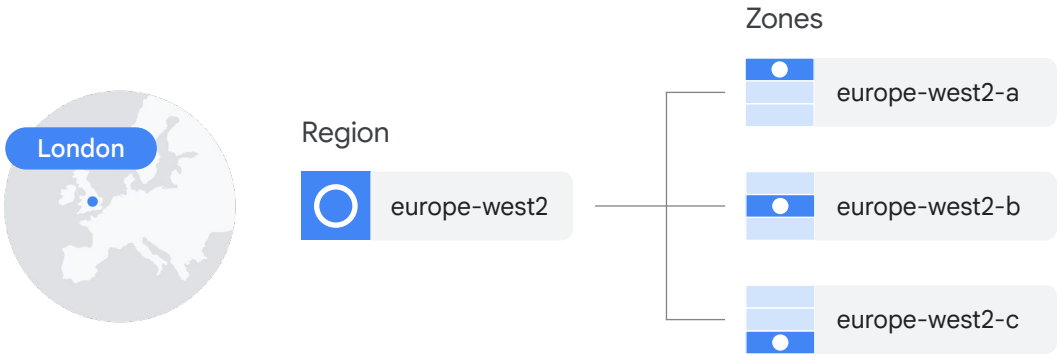
According to some publicly available estimates, Google's network carries as much as 40% of the world's internet traffic every day. Google's network is the largest network of its kind, and Google has invested billions of dollars over the years to build it.

This network is designed to give customers the highest possible throughput and lowest possible latencies for their applications by using more than 100 content caching nodes worldwide.

These are locations where high demand content is cached for quicker access, which allows applications to respond to user requests from the location that provides the quickest response time.

Google Cloud's infrastructure is based in seven major geographic locations: North America, South America, Europe, Africa, the Middle East, Asia, and Australia. Having multiple service locations is important because choosing where to locate applications affects qualities like availability, durability, and latency.

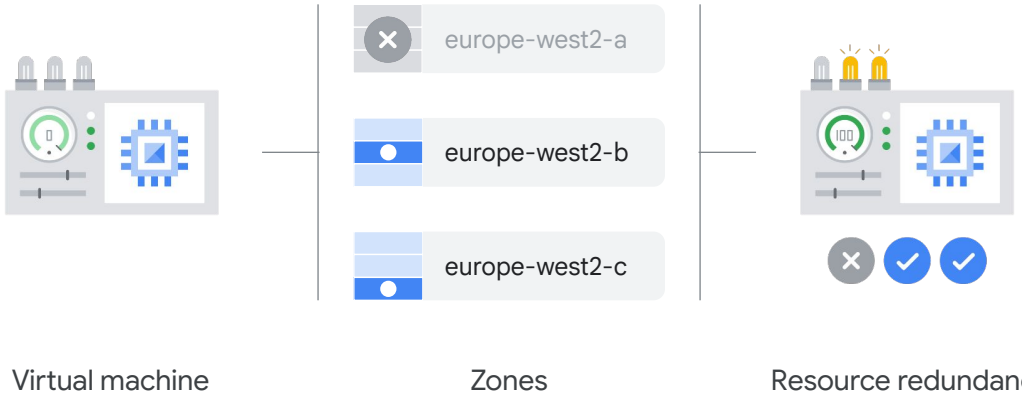
Locations, regions, and zones



Each of these locations is divided into several different **regions** and **zones**. Regions represent independent geographic areas and are composed of zones. For example, London, or europe-west2, is a region that currently comprises three different zones.

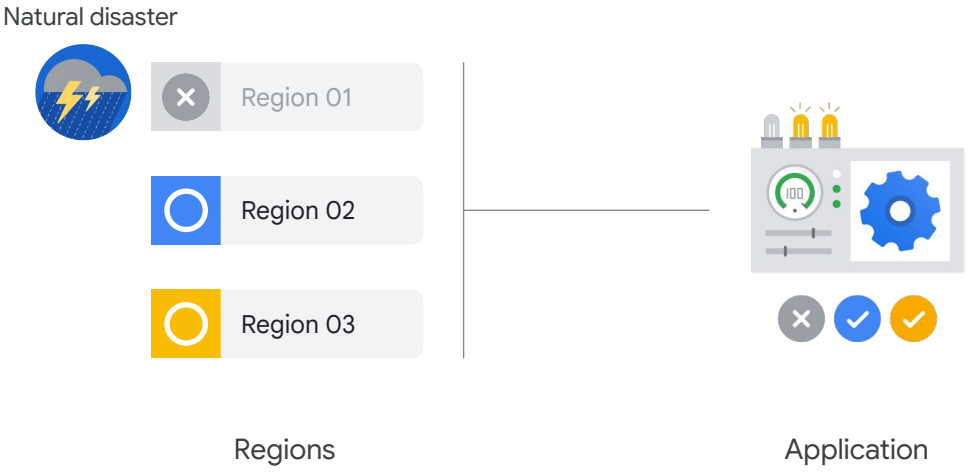
A zone is an area where Google Cloud resources are deployed.

Zones ensure resource redundancy



For example, if you use Compute Engine to launch a virtual machine, it will run in the zone that you specify to ensure resource redundancy.

Run resources in different zones



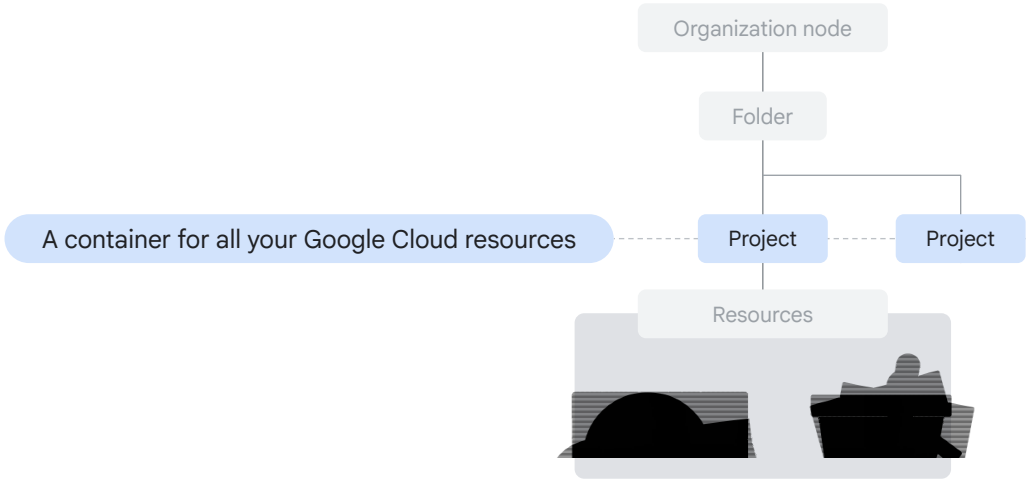
You can also run resources in different regions. This is useful for bringing applications closer to users around the world and also for protection if issues with an entire region occur, such as a natural disaster.

Google Cloud currently supports 124 zones in 41 regions, although this number is constantly increasing. You can find the most up-to-date numbers at cloud.google.com/about/locations.

Introduction to Google Cloud

Now let's explore how Google Cloud resources are structured and managed.

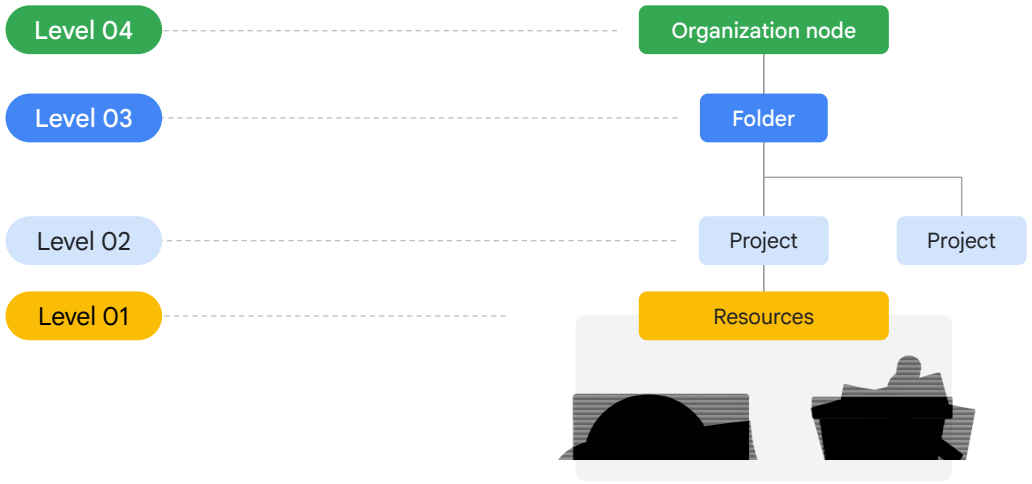
Every cloud resource you use must belong to a project



Every Google Cloud resource you use must belong to a project.

But what is a project? A project is a container for all your Google Cloud resources. It provides a way to organize resources, manage billing, and control access. Each project even has a unique identifier.

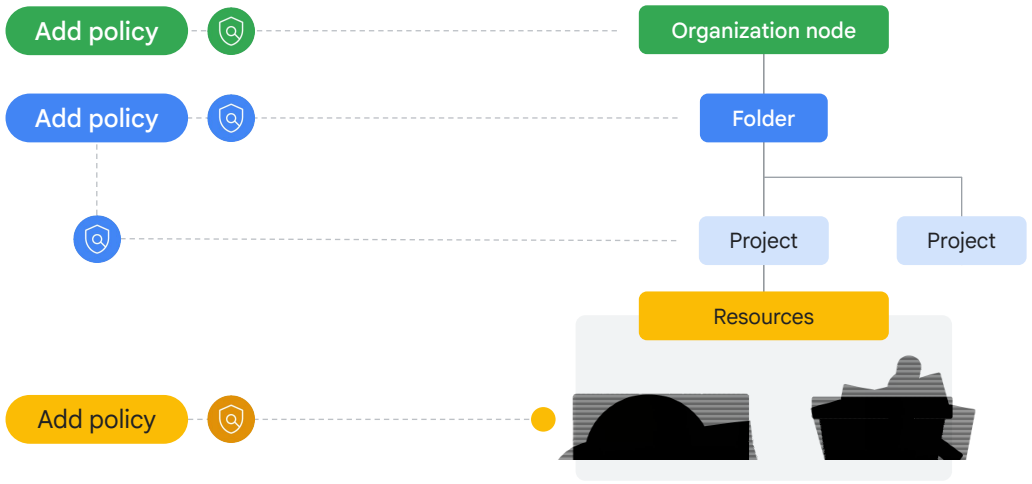
The Google Cloud resource hierarchy



Google Cloud's resource hierarchy contains four levels, and starting from the bottom up they are: resources, projects, folders, and an organization node.

1. **Resources** are at the first level. These represent containers, virtual machines, tables in BigQuery, or anything else in Google Cloud.
2. Resources are organized into **projects**, which sit on the second level.
3. Projects can be organized into **folders**, or even subfolders. These sit at the third level.
4. And then at the top level is an **organization node**, which encompasses all the projects, folders, and resources in your organization.

Policies can be defined at any level



It's important to understand this resource hierarchy, because it directly relates to how policies are managed and applied when you use Google Cloud.

Policies can be defined at the project, folder, and organization node levels. Some Google Cloud services allow policies and deny policies to be applied to individual resources, too.

Policies are inherited downward. This means that if you apply a policy to a folder, it will also apply to all of the projects within that folder.

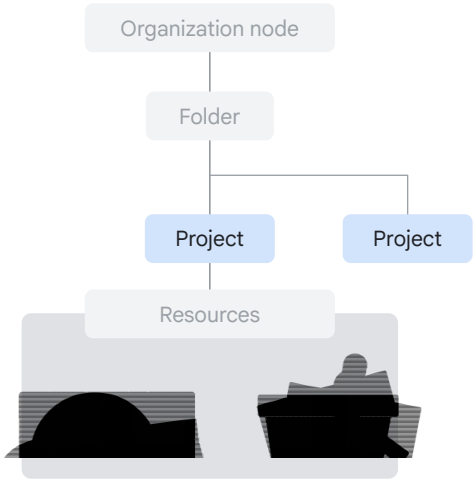
Projects are the basis for enabling and using services

01 Projects are separate entities under the organization node

02 Projects hold resources, each of which belongs to just one project

03 Projects can have different owners and users

04 Projects are billed and managed separately



Let's look at the second level of the resource hierarchy, projects, in a little more detail. Projects are the basis for enabling and using Google Cloud services, like managing APIs, enabling billing, adding and removing collaborators, and enabling other Google services.

Each project is a separate entity under the organization node, and each resource belongs to exactly one project. Projects can have different owners and users, because they're billed and managed separately.

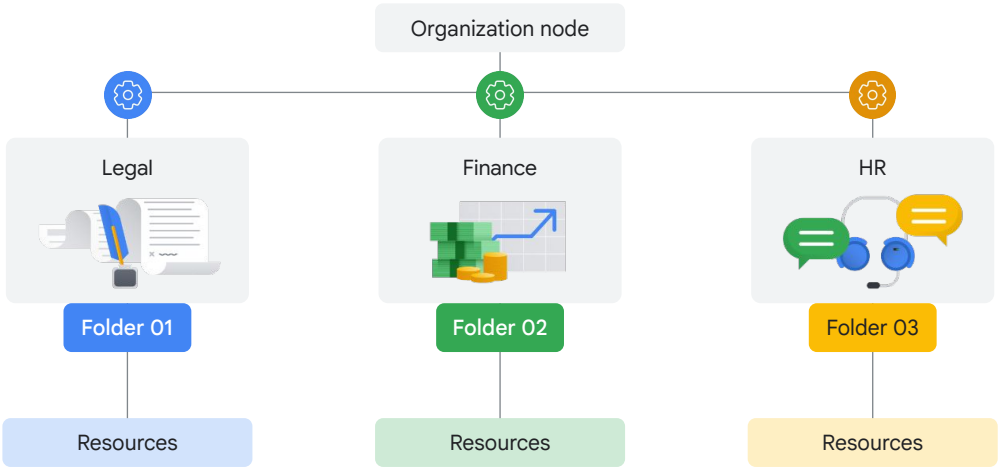
Project attributes

Project ID	Project name	Project number
Globally unique	Need not be unique	Globally unique
Assigned by Google Cloud but mutable during creation	Chosen by you	Assigned by Google Cloud
Immutable after creation	Mutable	Immutable

Each Google Cloud project has three identifying attributes: a project ID, a project name, and a project number.

- The **project ID** is a globally unique identifier assigned by Google that can't be changed after creation. They're what we refer to as being *immutable*. Project IDs are used in different contexts to inform Google Cloud of the exact project to work with.
- **Project names**, however, are user-created. They don't have to be unique, and they can be changed at any time, so they are mutable.
- Google Cloud also assigns each project a unique **project number**. It's helpful to know that these Google-generated numbers exist, but we won't explore them much in this course. They're mainly used internally by Google Cloud to keep track of resources.

Use folders to group projects

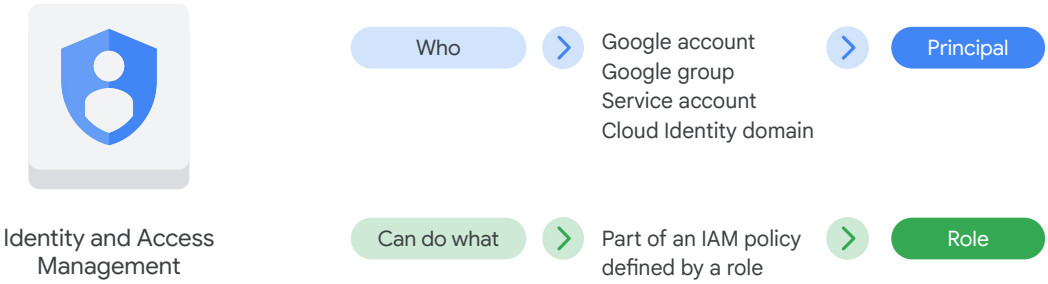


You can use folders to group projects under an organization in a hierarchy. For example, your organization might contain multiple departments, each with its own set of Google Cloud resources. Folders let you group these resources on a per-department basis.

Folders also give teams the ability to delegate administrative rights so they can work independently.

And when an organization node contains lots of folders, projects, and resources, a workforce might need to restrict who has access to what.

Use IAM to apply policies and give access to resources

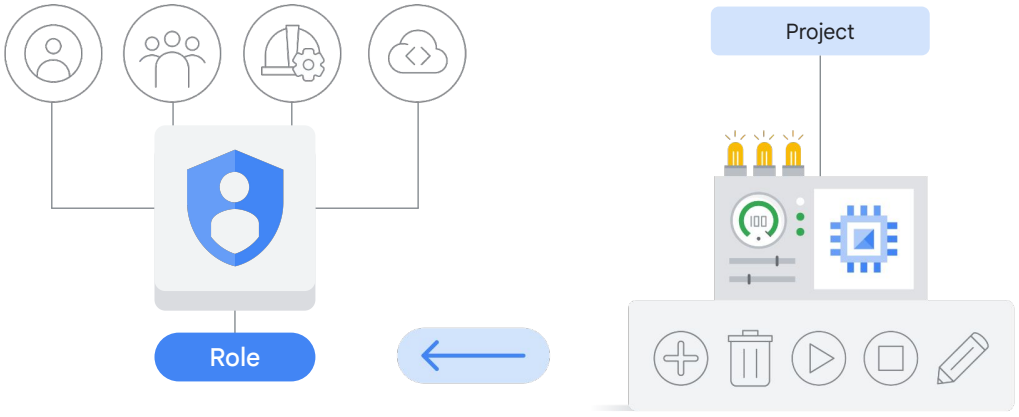


To help with this task, administrators can use **Identity and Access Management**, or IAM. With IAM, administrators can apply policies that define who can do what and on which resources.

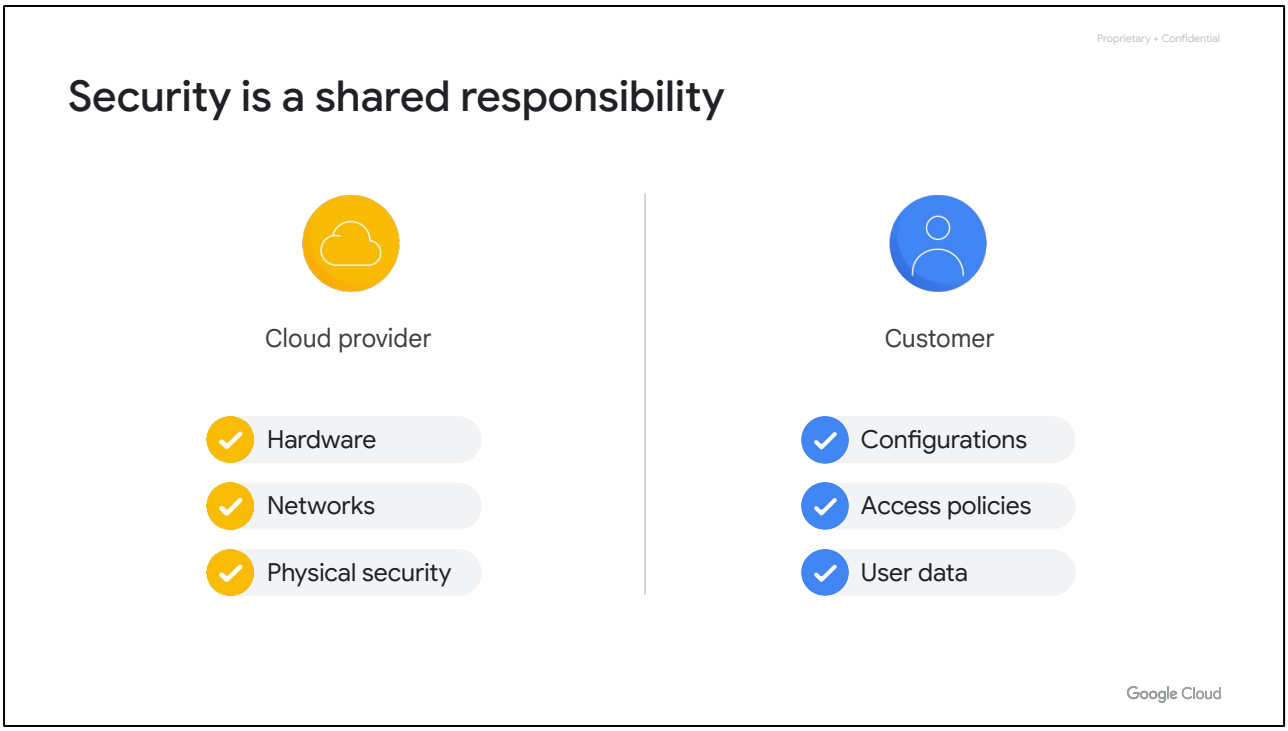
- The “who” part of an IAM policy can be a Google account, a Google group, a service account, or a Cloud Identity domain.
- A “who” is also called a “principal.” Each principal has its own identifier, usually an email address.
- The “can do what” part of an IAM policy is defined by a role. An IAM role is a collection of permissions.

When you grant a role to a principal, you grant all the permissions that the role contains.

Group permissions into a role



For example, to manage virtual machine instances in a project, you must be able to create, delete, start, stop and change virtual machines. So these permissions are grouped into a role to make them easier to understand and easier to manage.



So, who is responsible for security in the cloud? It's a shared responsibility between you and Google Cloud. A general guideline for shared responsibility is that "if you configure or store it, you're responsible for securing it."

This means that a **cloud provider** is responsible for securing the parts of the cloud that it directly controls, such as **hardware, networks, and physical security**.

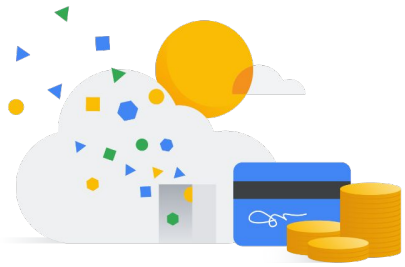
At the same time, the **customer** is responsible for securing anything that they create within the cloud, such as the **configurations, access policies, and user data**.

No matter which cloud provider you use, there is shared responsibility.

Introduction to Google Cloud

Let's explore how billing works in Google Cloud.

Billing



Billing is established at the project level.

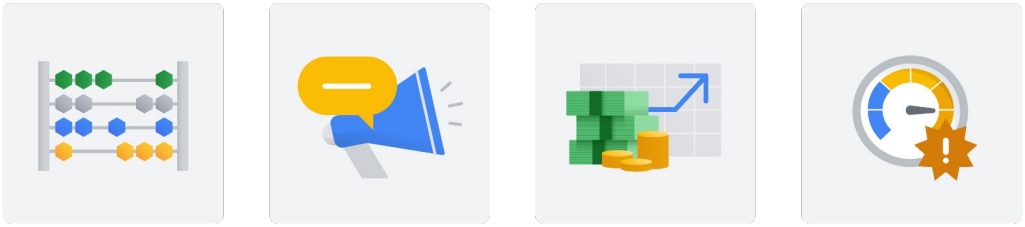
A billing account can be linked to zero or more projects.

Billing accounts are charged automatically and invoiced every month, or at every threshold limit.

Billing subaccounts can be used to separate billing by project.

- Billing is established at the project level of the Google Cloud resource hierarchy. This means that when you define a Google Cloud project, you link a billing account to it. This billing account is where you will configure all your billing information, including your payment option.
- A billing account can be linked to zero or more projects, but projects that aren't linked to a billing account can only use free Google Cloud services.
- Billing accounts are charged automatically and invoiced every month or at every threshold limit.
- Billing subaccounts can be used to separate billing by project. Some Google Cloud customers who resell Google Cloud services use subaccounts for each of their own clients.

Tools to help control costs



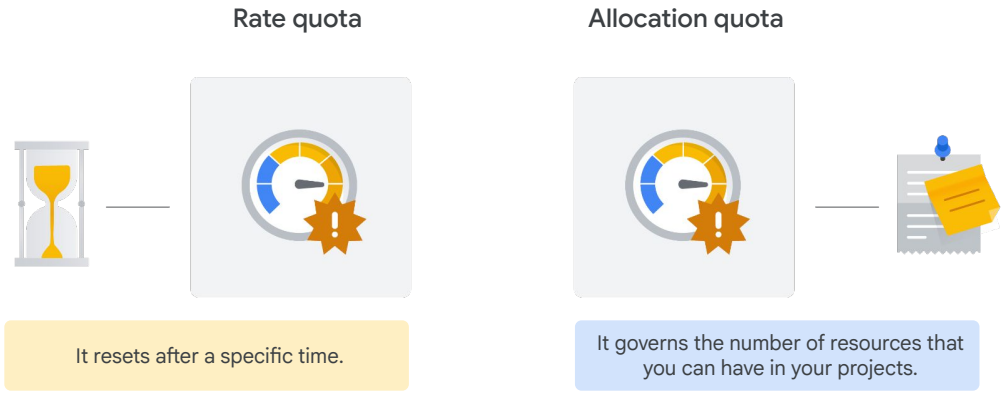
Budgets Alerts Reports Quotas

Now, you're probably thinking, "How can I ensure that I don't accidentally face a big Google Cloud bill?"

We provide a few tools to help.

- You can define **budgets** at the billing account level or at the project level. A budget can be a fixed limit, or it can be tied to another metric; for example, a percentage of the previous month's spend.
- To be notified when costs approach your budget limit, you can create an **alert**. For example, with a budget limit of $20,000 and an alert set at 90%, you'll receive a notification alert when your expenses reach $18,000. Alerts are generally set at 50%, 90% and 100%, but can also be customized.
- **Reports** is a visual tool in the Google Cloud console that lets you monitor expenditure based on a project or services.
- Finally, Google Cloud also implements **quotas**, which are designed to prevent the over-consumption of resources because of an error or a malicious attack. This way both account owners and the Google Cloud community as a whole are protected.

Quotas



There are two types of quotas: rate quotas and allocation quotas. Both are applied at the project level.

- **Rate quotas** reset after a specific time. For example, by default, the GKE service implements a quota of 3,000 calls to its API from each Google Cloud project every 100 seconds. After that 100 seconds, the limit is reset.
- **Allocation quotas** govern the number of resources that you can have in your projects. For example, by default, each Google Cloud project has a quota that allows it no more than 15 Virtual Private Cloud networks.

Although all projects start with the same quotas, you can change some of them by requesting an increase from Google Cloud support.

Introduction to Google Cloud

01 Google Cloud compute offerings

02 Google's global network

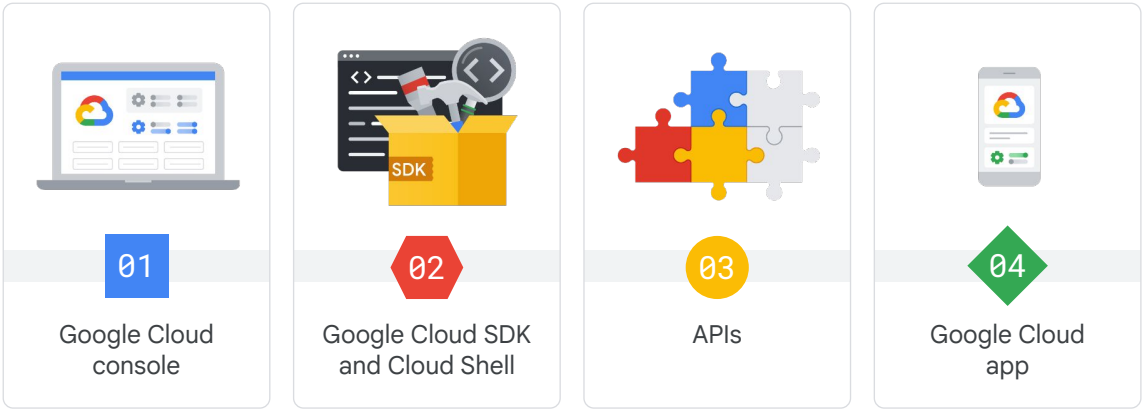
03 Managing cloud resources

04 Billing tools

05 Four ways to interact with Google Cloud

Now that you explored how resources in Google Cloud run, are organized, and billed to you, it's time to see how you actually interact with Google Cloud to start your work.

Ways to access and interact with Google Cloud



You can use four Google products to access and interact with Google Cloud. The **Google Cloud console**, the **Google Cloud SDK** and **Cloud Shell**, the **APIs**, and the **Google Cloud app**.

Let's explore each.

Google Cloud console

console.cloud.google.com

- It's a simple web-based graphical user interface.
- You can easily find resources, check their health, have full management control over them, and set budgets.
- It provides a search facility to quickly find resources and connect to instances through SSH in the browser.

The **first** is the **Google Cloud console**, which is Google Cloud's graphical user interface, or GUI, and it helps you deploy, scale, and diagnose production issues in a simple web-based interface.

With the Google Cloud console, you can easily find your resources, check their health, have full management control over them, and set budgets to control how much you spend on them.

The Google Cloud console also provides a search facility to quickly find resources and connect to instances through SSH in the browser.

The Google Cloud console is available from console.cloud.google.com. You'll get some experience with the console during an upcoming lab.

Google Cloud SDK and Cloud Shell

- It's a set of tools to manage resources and applications hosted on Google Cloud.
- The **gcloud tool** provides the main command-line interface for Google Cloud products and services.
- **gcloud storage** provides access to Cloud Storage from the command line.
- **bq** is command-line tool for BigQuery.

The **second** products are the Google Cloud SDK and Cloud Shell. Unlike the Google Cloud console, you can download and install the Google Cloud SDK locally onto a computer.

The **Google Cloud SDK** is a set of tools you can use to manage resources and applications hosted on Google Cloud. These include the *gcloud tool*, which provides the main command-line interface for Google Cloud products and services, *gcloud storage*, which lets you access Cloud Storage from the command line, and *bq*, a command-line tool for BigQuery.

When installed, all of the tools within the Google Cloud SDK are located under the *bin* directory.

Google Cloud SDK and Cloud Shell

- It provides command-line access to cloud resources directly from a browser.
- It's a Debian-based virtual machine with a persistent 5-GB home directory.
- Each Cloud Shell VM is ephemeral.
- it provides web preview functionality and built-in authorization for access to console projects and resources, including your GKE resources.
- The Google Cloud SDK gcloud command and other utilities are always installed, available, up to date, and fully authenticated.

Cloud Shell provides command-line access to cloud resources directly from a browser.

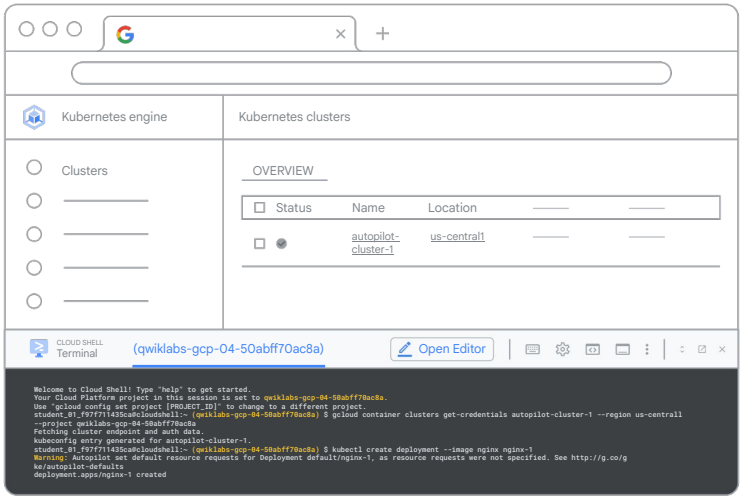
Cloud Shell is a Debian-based virtual machine with a persistent 5 gigabyte home directory, which makes it easy to manage Google Cloud projects and resources.

Each Cloud Shell VM is ephemeral, which means that it will be stopped whenever you stop using it interactively, and it'll be restarted when you re-enter Cloud Shell.

It also provides web preview functionality and built-in authorization for access to console projects and resources, including your GKE resources.

With Cloud Shell, the Google Cloud SDK gcloud command and other utilities are always installed, available, up to date, and fully authenticated.

Google Cloud SDK and Cloud Shell



The console's GKE area has a web-based interface for administering GKE resources.

The Cloud Shell is the place to launch commands to administer those GKE resources. Some of those commands are from the Google Cloud SDK, and others will be specific to your workload. Later in this course, you learn about the kubectl command, and you can see it being launched from Cloud Shell here.

APIs

- The services that comprise Google Cloud offer APIs so that the code you write can control them.
- The Google APIs Explorer shows which APIs are available, and in which versions.
- Developers often use APIs to build applications that allocate and manage Google Cloud resources.
- Our present focus is on letting Kubernetes manage resources for us.

The **third** way to access Google Cloud is through application programming interfaces, or **APIs**.

The services that comprise Google Cloud offer **APIs** so that the code you write can control them. The Google Cloud console includes a tool called the *Google APIs Explorer* that shows which APIs are available, and in which versions. You can try these APIs interactively, even those that require user authentication.

One important point to note is that developers often use APIs to build applications that allocate and manage Google Cloud resources. However, our present focus is on letting Kubernetes manage resources for us.

Google Cloud app

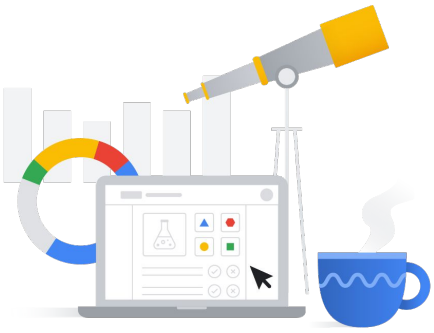
- Start, stop, and use SSH to connect to Compute Engine instances and see logs from each instance.
- Stop and start Cloud SQL instances.
- Administer applications deployed on App Engine by viewing errors, rolling back deployments, and changing traffic splitting.

And finally, the **fourth** way to access and interact with Google Cloud is with the **Google Cloud app**, which can be used to start, stop, and use SSH to connect to Compute Engine instances and see logs from each instance.

It also lets you stop and start Cloud SQL instances. Additionally, you can administer applications deployed on App Engine by viewing errors, rolling back deployments, and changing traffic splitting.

The Google Cloud app isn't relevant for the purposes of this course.

Let's get some practice



It's time to gain some hands-on experience with some of the Google Cloud tools featured in this section of the course. We'll use Qwiklabs.

Lab environment

For each lab, Qwiklabs offers:

- A free set of resources for a fixed amount of time
- A clean environment with permissions

Qwiklabs provisions you with Google account credentials, so you can access the Google Cloud console for each lab at no cost.

Specifically, for each lab, Qwiklabs offers:

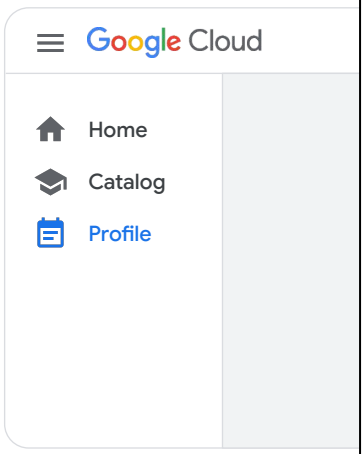
- A free set of resources for a fixed amount of time
- A clean environment with permissions

Open Qwiklabs

1. **Open an incognito window** (or private/anonymous window).
2. **Go** to the Qwiklabs URL your instructor provides.
3. **Sign in** with existing account or **join** with new account (with the email you used to register for the course).
4. Launch the course from **Profile**.

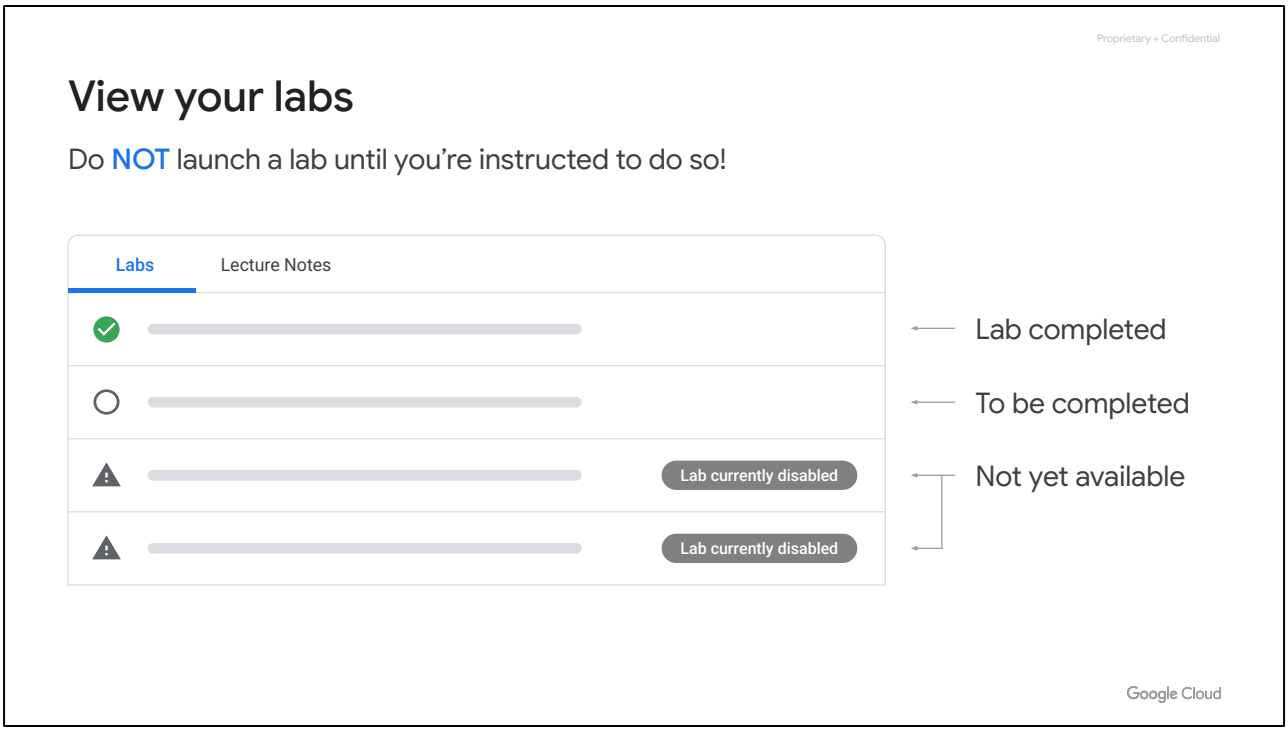
Access issues

The process to open Qwiklabs can differ based on the credentials used. Reach out to your trainer if you have any access issues.



Go ahead and open Qwiklabs:

1. **Open an incognito window** (or private/anonymous window). Using an incognito browser window reduces the risk that you will accidentally do the labs by using your own Google Cloud account instead of Qwiklabs.
2. **Go** to the Qwiklabs URL your instructor provides.
3. **Sign in** with an existing account or **join** with a new account (with the email you used to register for the course).
4. Launch the course from **Profile**.



After you launch the course, you can view your labs. The lab list will indicate whether a lab is:

- Completed (by you)
- To be completed
- Not yet available

Your instructor will let you know when it's time to launch a lab. When you start a lab, you can't pause and restart it, so you need a continuous block of time to complete the work.

Lab: Accessing the Google Cloud console and Cloud Shell

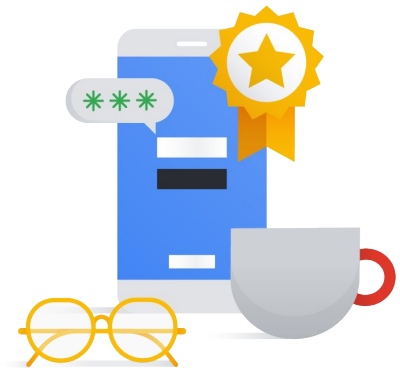
01 Creating buckets from the Google Cloud console and Cloud Shell.

02 Creating virtual machines from the Google Cloud console and Cloud Shell.

03 Creating service accounts from the Google Cloud console and Cloud Shell

04 Executing other commands through Cloud Shell

In the lab titled “Accessing the Google Cloud console and Cloud Shell,” you’ll get practice creating buckets, virtual machines, and service accounts from both the Google Cloud console and Cloud Shell. You’ll also get practice executing other commands through Cloud Shell.



Time for a short quiz

Let’s pause for a short quiz.

Quiz | Question 1 of 4

Question

You are considering deploying a solution by using containers on Google Cloud. What Google Cloud solution provides a managed compute platform with native support for containers?

A. Compute Engine autoscaling groups

B. Google Kubernetes Engine clusters

C. Cloud Run functions

D. Artifact Registry

Quiz | Question 1 of 4

Answer

You are considering deploying a solution by using containers on Google Cloud. What Google Cloud solution provides a managed compute platform with native support for containers?

A. Compute Engine autoscaling groups

B. Google Kubernetes Engine clusters 

C. Cloud Run functions

D. Artifact Registry

Quiz | Question 2 of 4

Question

What Identity and Access Management (IAM) hierarchy structure is best for building an application in Google Cloud?

A. Create a new folder inside your organization node, then create projects inside that folder for the resources.

B. Create new projects for each of the component applications, then create folders inside those for the resources.

C. Create a new organization node for the project, then create all projects and resources inside the new organization node.

D. Create new projects and resources inside departmental folders for the resources needed, organized by the component applications.

Quiz | Question 2 of 4

Answer

What Identity and Access Management (IAM) hierarchy structure is best for building an application in Google Cloud?

A. Create a new folder inside your organization node, then create projects inside that folder for the resources.

B. Create new projects for each of the component applications, then create folders inside those for the resources.

C. Create a new organization node for the project, then create all projects and resources inside the new organization node.

D. Create new projects and resources inside departmental folders for the resources needed, organized by the component applications.

Quiz | Question 3 of 4

Question

You are developing a new product for a customer and need to be mindful of cost and resources. What Google Cloud tools can be used to ensure that costs stay manageable before consumption gets too high?

A. Configure the billing account for each project associated with the product.

B. Configure the billing account at the product folder level in the resource hierarchy.

C. Configure quotas and limits for each product folder.

D. Set up budgets and alerts at the project level.

Quiz | Question 3 of 4

Answer

You are developing a new product for a customer and need to be mindful of cost and resources. What Google Cloud tools can be used to ensure that costs stay manageable before consumption gets too high?

A. Configure the billing account for each project associated with the product.

B. Configure the billing account at the product folder level in the resource hierarchy.

C. Configure quotas and limits for each product folder.

D. Set up budgets and alerts at the project level. 

Quiz | Question 4 of 4

Question

One of the main characteristics of cloud computing is that resources are elastic. What does that mean?

A. You can quickly get more resources when you need them.

B. When customers need more resources, they can get more. When they need less, they can scale back.

C. You share resources from a large pool that enables economies of scale.

D. Resources can be allocated automatically.

Quiz | Question 4 of 4

Answer

One of the main characteristics of cloud computing is that resources are elastic. What does that mean?

A. You can quickly get more resources when you need them.

B. When customers need more resources, they can get more. When they need less, they can scale back.

C. You share resources from a large pool that enables economies of scale.

D. Resources can be allocated automatically.